



(12) 发明专利申请

(10) 申请公布号 CN 105700931 A

(43) 申请公布日 2016. 06. 22

(21) 申请号 201610099126. 8

(22) 申请日 2016. 02. 23

(71) 申请人 北京蓝海讯通科技股份有限公司

地址 100192 北京市海淀区西小口路 66 号
中关村东升科技园 A-5 号楼三、四层

(72) 发明人 何晓阳 洪森 王美娟

(74) 专利代理机构 北京集佳知识产权代理有限公司 11227

代理人 王宝筠

(51) Int. Cl.

G06F 9/445(2006. 01)

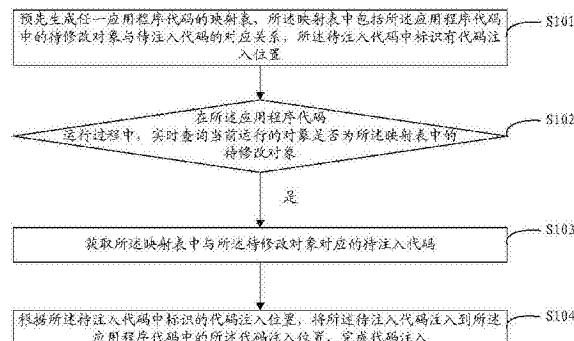
权利要求书2页 说明书5页 附图1页

(54) 发明名称

一种代码注入方法及装置

(57) 摘要

本发明实施例公开了一种代码注入方法及装置，所述方法包括：预先生成任一应用程序代码的映射表，所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系，所述待注入代码中标识有代码注入位置；在所述应用程序代码运行过程中，实时查询当前运行的对象是否为所述映射表中的待修改对象，如果是，则获取所述映射表中与所述待修改对象对应的待注入代码；根据所述待注入代码中标识的代码注入位置，将所述待注入代码注入到所述应用程序代码中的所述代码注入位置，完成代码注入。本发明能够自动化完成代码注入。



1.一种代码注入方法,其特征在于,所述方法包括:

预先生成任一应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置;

在所述应用程序代码运行过程中,实时查询当前运行的对象是否为所述映射表中的待修改对象,如果是,则获取所述映射表中与所述待修改对象对应的待注入代码;

根据所述待注入代码中标识的代码注入位置,将所述待注入代码注入到所述应用程序代码中的所述代码注入位置,完成代码注入。

2.根据权利要求1所述的方法,其特征在于,所述预先生成任一应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置,包括:

接收预先对应用程序代码中对象的注解,所述注解用于标识待修改对象、待注入代码以及代码注入位置;

根据所述应用程序代码中的注解,生成所述应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置。

3.根据权利要求1所述的方法,其特征在于,所述待修改对象包括java代码中的类、方法。

4.根据权利要求1所述的方法,其特征在于,所述根据所述待注入代码中标识的代码注入位置,将所述待注入代码注入到所述应用程序代码中的所述代码注入位置,完成代码注入,包括:

获取所述应用程序代码中所述待修改对象对应的字节码;

根据所述待注入代码中标识的代码注入位置,将所述映射表中所述待修改对象对应的待注入代码转化为字节码后,与所述待修改对象对应的字节码合并,完成代码注入。

5.根据权利要求1所述的方法,其特征在于,所述方法还包括:

如果当前运行的对象不是所述映射表中的待修改对象,则继续执行所述应用程序代码的下一个对象。

6.一种代码注入装置,其特征在于,所述装置包括:

生成模块,用于预先生成任一应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置;

查询模块,用于在所述应用程序代码运行过程中,实时查询当前运行的对象是否为所述映射表中的待修改对象;

获取模块,用于在所述查询模块的查询结果为是时,获取所述映射表中与所述待修改对象对应的待注入代码;

注入模块,用于根据所述待注入代码中标识的代码注入位置,将所述待注入代码注入到所述应用程序代码中的所述代码注入位置,完成代码注入。

7.根据权利要求6所述的模块,其特征在于,所述生成模块,包括:

接收子模块,用于接收预先对应用程序代码中对象的注解,所述注解用于标识待修改对象、待注入代码以及代码注入位置;

生成子模块，用于根据所述应用程序代码中的注解，生成所述应用程序代码的映射表，所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系，所述待注入代码中标识有代码注入位置。

8. 根据权利要求6所述的装置，其特征在于，所述待修改对象包括java代码中的类、方法。

9. 根据权利要求6所述的装置，其特征在于，所述注入模块，包括：

获取子模块，用于获取所述应用程序代码中所述待修改对象对应的字节码；

合并子模块，用于根据所述待注入代码中标识的代码注入位置，将所述映射表中所述待修改对象对应的待注入代码转化为字节码后，与所述待修改对象对应的字节码合并，完成代码注入。

10. 根据权利要求6所述的模块，其特征在于，所述装置还包括：

执行模块，用于在所述查询模块的查询结果为否时，继续执行所述应用程序代码的下一个对象。

一种代码注入方法及装置

技术领域

[0001] 本发明涉及数据处理领域,具体涉及一种代码注入方法及装置。

背景技术

[0002] 代码注入是将实现一定功能的代码段注入至应用程序代码中,以增加原有应用程序能够实现的功能。由于实现代码注入实质上是在应用程序运行的过程中,动态的将代码段插入到预设位置的过程。

[0003] 目前,代码注入过程是靠人工完成的,即将待注入代码人工的插入至应用程序代码中,完成代码的注入过程。但是,这对开发人员的专业知识要求较高,且人工完成代码注入效率较低。

发明内容

[0004] 有鉴于此,本发明提供了一种代码注入方法,能够自动化的完成代码注入。

[0005] 本发明提供了一种代码注入方法,所述方法包括:

[0006] 预先生成任一应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置;

[0007] 在所述应用程序代码运行过程中,实时查询当前运行的对象是否为所述映射表中的待修改对象,如果是,则获取所述映射表中与所述待修改对象对应的待注入代码;

[0008] 根据所述待注入代码中标识的代码注入位置,将所述待注入代码注入到所述应用程序代码中的所述代码注入位置,完成代码注入。

[0009] 优选地,所述预先生成任一应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置,包括:

[0010] 接收预先对应用程序代码中对象的注解,所述注解用于标识待修改对象、待注入代码以及代码注入位置;

[0011] 根据所述应用程序代码中的注解,生成所述应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置。

[0012] 优选地,所述待修改对象包括java代码中的类、方法。

[0013] 优选地,所述根据所述待注入代码中标识的代码注入位置,将所述待注入代码注入到所述应用程序代码中的所述代码注入位置,完成代码注入,包括:

[0014] 获取所述应用程序代码中所述待修改对象对应的字节码;

[0015] 根据所述待注入代码中标识的代码注入位置,将所述映射表中所述待修改对象对应的待注入代码转化为字节码后,与所述待修改对象对应的字节码合并,完成代码注入。

[0016] 优选地,所述方法还包括:

[0017] 如果当前运行的对象不是所述映射表中的待修改对象,则继续执行所述应用程序

代码的下一个对象。

[0018] 本发明还提供了一种代码注入装置,所述装置包括:

[0019] 生成模块,用于预生成任一应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置;

[0020] 查询模块,用于在所述应用程序代码运行过程中,实时查询当前运行的对象是否为所述映射表中的待修改对象;

[0021] 获取模块,用于在所述查询模块的查询结果为是时,获取所述映射表中与所述待修改对象对应的待注入代码;

[0022] 注入模块,用于根据所述待注入代码中标识的代码注入位置,将所述待注入代码注入到所述应用程序代码中的所述代码注入位置,完成代码注入。

[0023] 优选地,所述生成模块,包括:

[0024] 接收子模块,用于接收预先对应用程序代码中对象的注解,所述注解用于标识待修改对象、待注入代码以及代码注入位置;

[0025] 生成子模块,用于根据所述应用程序代码中的注解,生成所述应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置。

[0026] 优选地,所述待修改对象包括java代码中的类、方法。

[0027] 优选地,所述注入模块,包括:

[0028] 获取子模块,用于获取所述应用程序代码中所述待修改对象对应的字节码;

[0029] 合并子模块,用于根据所述待注入代码中标识的代码注入位置,将所述映射表中所述待修改对象对应的待注入代码转化为字节码后,与所述待修改对象对应的字节码合并,完成代码注入。

[0030] 优选地,所述装置还包括:

[0031] 执行模块,用于在所述查询模块的查询结果为否时,继续执行所述应用程序代码的下一个对象。

[0032] 本发明提供了一种代码注入方法,其中,预生成任一应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置。其次,在所述应用程序代码运行过程中,实时查询当前运行的对象是否为所述映射表中的待修改对象,如果是,则获取所述映射表中与所述待修改对象对应的待注入代码。最后,根据所述待注入代码中标识的代码注入位置,将所述待注入代码注入到所述应用程序代码中的所述代码注入位置,完成代码注入。本发明提供的代码注入方法能够自动化的完成代码注入。

附图说明

[0033] 为了更清楚地说明本申请实施例中的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

- [0034] 图1为本发明实施例提供的一种代码注入方法流程图；
- [0035] 图2为本发明实施例提供的一种代码注入装置的结构示意图；
- [0036] 图3为本发明实施例提供的一种生成模块的结构示意图。

具体实施方式

[0037] 下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

[0038] 本发明实施例提供了一种代码注入方法，参考图1，为本发明实施例提供的一种代码注入方法流程图，所述方法包括：

[0039] S101：预先生成任一应用程序代码的映射表，所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系，所述待注入代码中标识有代码注入位置。

[0040] 本发明实施例中，在为任一应用程序进行代码注入之前，首先生成所述应用程序代码的映射表。其中，所述映射表中包括该应用程序代码中待修改对象，以及对应的待注入代码，所述映射表的格式可以以map的形式实现。另外，所述待注入代码中标识有代码注入位置，即为所述待注入代码插入的原代码中的位置。

[0041] 实际应用中，本实施例可以预先将用于生成映射表的探针部署到用户应用程序中，一般部署在premain()函数中。本发明实施例可以基于注解的方式实现，具体的，预先对所述应用程序代码中的对象进行注解，所述注解用于标识待修改对象、待注入代码以及代码注入位置。其次，在执行premain()函数时，利用所述探针，根据所述应用程序代码中的注解，生成所述应用程序代码的映射表，其中所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系，所述待注入代码中标识有代码注入位置。

[0042] S102：在所述应用程序代码运行过程中，实时查询当前运行的对象是否为所述映射表中的待修改对象，如果是，则执行S103。

[0043] S103：获取所述映射表中与所述待修改对象对应的待注入代码。

[0044] 本发明实施例中，所述映射表中的待修改对象可以包括java代码中的类、方法等。在所述应用程序代码运行的过程中，即java虚拟机加载代码中的任一个类或方法时，实时查询预生成的映射表，以确定当前运行的对象是否为所述映射表中的待修改对象。

[0045] 如果系统在所述映射表中查询到当前运行的对象属于待修改对象，则从所述映射表中获取所述待修改对象对应的待注入代码。

[0046] 实际应用中，在系统执行main()函数后，实时查询当前运行的对象是否为所述映射表中的待修改对象。

[0047] 另外，如果系统在所述映射表中未查询到当前运行的对象，则系统继续执行所述应用程序代码的下一个对象即可。

[0048] S104：根据所述待注入代码中标识的代码注入位置，将所述待注入代码注入到所述应用程序代码中的所述代码注入位置，完成代码注入。

[0049] 本发明实施例中，在所述映射表中获取所述待修改对象对应的待注入代码后，获取所述待注入代码中标识的代码注入位置。根据所述代码注入位置，将所述待注入代码注

入至所述应用程序代码中,完成代码注入。本发明实施例注入的代码可以用于监控用户应用程序的性能以及方法调用信息等,实现用户应用程序性能监控。

[0050] 实际应用中,所述应用程序中代码是以字节码的形式运行于java虚拟机中的,本发明实施例可以借助于google开发的ASM工具完成应用程序中字节码的修改。

[0051] 实际操作中,本发明实施例首先获取所述应用程序代码中所述待修改对象对应的字节码,即所述待修改对象的原代码。其次,将所述映射表中所述待修改对象对应的待注入代码转化为字节码。最后,根据所述待注入代码中标识的代码注入位置,将转化后的字节码与所述应用程序代码中所述待修改对象对应的字节码合并,完成代码注入。

[0052] 由于只有字节码格式才能被直接插入到运行于java虚拟机的应用程序代码中,所以,本实施例在将映射表中的待注入代码注入的过程中,将所述待注入代码转化为字节码运行于java虚拟机中。

[0053] 由于本发明实施例的字节码转化过程是自动化完成的,与现有的代码人工注入方法相比,不需要人工将代码段翻译成字节码。也就是说,本发明实施例不要求技术开发人员要有很好的技术背景,不需要具备将代码段人工翻译成字节码的专业能力。

[0054] 本发明实施例提供的代码注入方法中,预生成任一应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置。其次,在所述应用程序代码运行过程中,实时查询当前运行的对象是否为所述映射表中的待修改对象,如果是,则获取所述映射表中与所述待修改对象对应的待注入代码。最后,根据所述待注入代码中标识的代码注入位置,将所述待注入代码注入到所述应用程序代码中的所述代码注入位置,完成代码注入。本发明实施例提供的代码注入方法能够自动化的完成代码注入。

[0055] 本发明实施例还提供了一种代码注入装置,参考图2,图2为本发明实施例提供的一种代码注入装置的结构示意图,所述装置包括:

[0056] 生成模块201,用于预生成任一应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置;

[0057] 查询模块202,用于在所述应用程序代码运行过程中,实时查询当前运行的对象是否为所述映射表中的待修改对象;

[0058] 获取模块203,用于在所述查询模块的查询结果为是时,获取所述映射表中与所述待修改对象对应的待注入代码;

[0059] 注入模块204,用于根据所述待注入代码中标识的代码注入位置,将所述待注入代码注入到所述应用程序代码中的所述代码注入位置,完成代码注入。

[0060] 其中,参考图3,为本发明实施例提供的生成模块201的结构示意图,所述生成模块201,包括:

[0061] 接收子模块301,用于接收预先对应用程序代码中对象的注解,所述注解用于标识待修改对象、待注入代码以及代码注入位置;

[0062] 生成子模块302,用于根据所述应用程序代码中的注解,生成所述应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置。

- [0063] 实际应用中,所述待修改对象包括java代码中的类、方法。
- [0064] 实际应用中,所述注入模块204,包括:
- [0065] 获取子模块,用于获取所述应用程序代码中所述待修改对象对应的字节码;
- [0066] 合并子模块,用于根据所述待注入代码中标识的代码注入位置,将所述映射表中所述待修改对象对应的待注入代码转化为字节码后,与所述待修改对象对应的字节码合并,完成代码注入。
- [0067] 具体的,所述装置还包括:
- [0068] 执行模块,用于在所述查询模块的查询结果为否时,继续执行所述应用程序代码的下一个对象。
- [0069] 本发明实施例提供的代码注入装置中,生成模块,用于预先生成任一应用程序代码的映射表,所述映射表中包括所述应用程序代码中的待修改对象与待注入代码的对应关系,所述待注入代码中标识有代码注入位置。查询模块,用于在所述应用程序代码运行过程中,实时查询当前运行的对象是否为所述映射表中的待修改对象。获取模块,用于在所述查询模块的查询结果为是时,获取所述映射表中与所述待修改对象对应的待注入代码。注入模块,用于根据所述待注入代码中标识的代码注入位置,将所述待注入代码注入到所述应用程序代码中的所述代码注入位置,完成代码注入。利用本发明实施例提供的代码注入装置能够自动化完成代码注入。
- [0070] 对于装置实施例而言,由于其基本对应于方法实施例,所以相关之处参见方法实施例的部分说明即可。以上所描述的装置实施例仅仅是示意性的,其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。本领域普通技术人员在不付出创造性劳动的情况下,即可以理解并实施。
- [0071] 需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。
- [0072] 以上对本发明实施例所提供的一种代码注入方法及装置进行了详细介绍,本文中应用了具体个例对本发明的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本发明的方法及其核心思想;同时,对于本领域的一般技术人员,依据本发明的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本发明的限制。

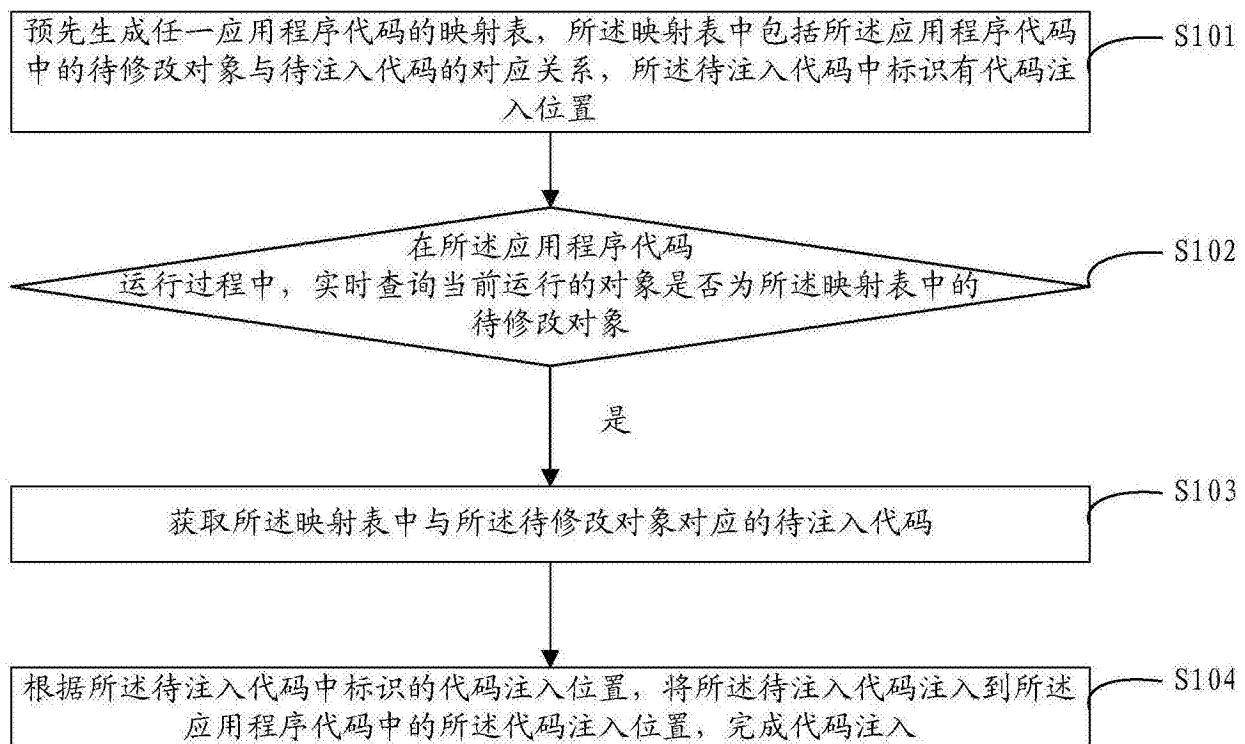


图1

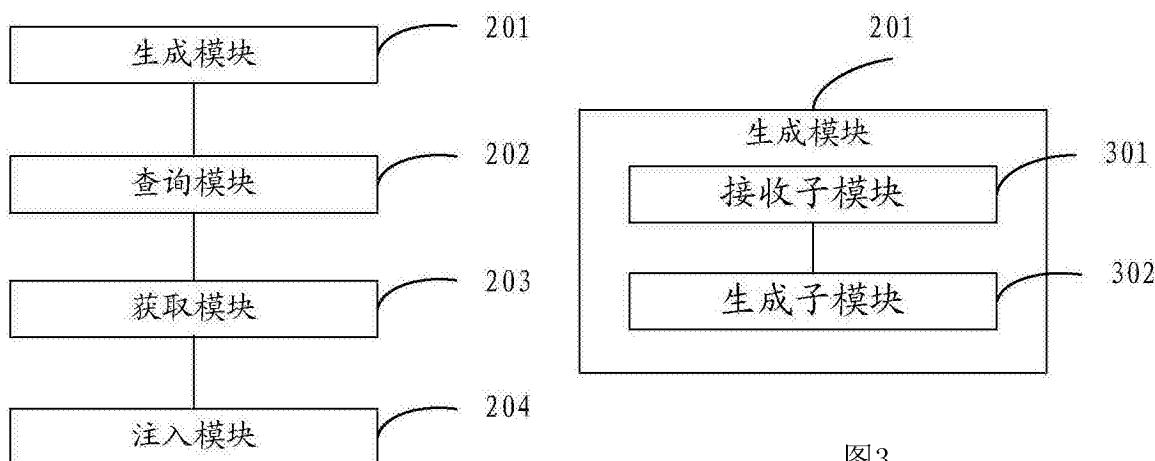


图3

图2